Package ‘hglm’

February 20, 2015

Type Package
Title Hierarchical Generalized Linear Models
Version 2.0-11
Date 2014-10-30
Author Xia Shen, Moudud Alam, Lars Ronnegard
Maintainer Xia Shen <xia.shen@ki.se>
Description This package fits hierarchical generalized linear models. It can be used for linear mixed models and generalized linear mixed models with random effects for a variety of links and a variety of distributions for both the outcomes and the random effects. Fixed effects can also be fitted in the dispersion part of the mean model.

BugReports https://r-forge.r-project.org/tracker/?group_id=558
License GPL (>= 2)
LazyLoad yes
Depends R (>= 2.10), utils, Matrix, MASS, hglm.data
NeedsCompilation no
Repository CRAN
Date/Publication 2014-10-30 17:07:51

R topics documented:

  hglm-package .................................................. 2
  Beta .............................................................. 3
  CAR .............................................................. 3
  hglm ............................................................. 4
  hglm2 ............................................................ 9
  lrt ............................................................... 14
  plot.hglm ....................................................... 15
  SAR ............................................................. 17

Index 19
Description

The hglm package is used to fit hierarchical generalized linear models. It can be used for linear mixed models and generalized linear models with random effects for a variety of links and a variety of distributions for both the outcomes and the random effects. Fixed effects can also be fitted in the dispersion part of the model. The function can be called either by specifying the design matrices or as a `formula`. The default estimation method is extended quasi likelihood (EQL; Lee et al., 2006) but from version 2.0 the EQL1 correction has been implemented as well.

Details

Package: hglm
Type: Package
Version: 2.0-11
Date: 2014-10-30
Discussion: https://r-forge.r-project.org/forum/?group_id=558
BugReports: https://r-forge.r-project.org/tracker/?group_id=558
License: GPL (>= 2)
LazyLoad: yes
Depends: R (>= 2.10), utils, Matrix, MASS, hglm.data

Author(s)

Xia Shen, Moudud Alam, Lars Ronnegard

Maintainer: Xia Shen <xia.shen@ki.se>

References


Beta

Moudud Alam, Lars Ronnegard, Xia Shen (2014). **Fitting conditional and simultaneous autoregressive spatial models in hglm. Submitted.**

See Also

* hglm, hglm2, plot.hglm*

---

Beta  
*Extended Beta Family*

**Description**

A function used in the hglm package which extends the usage of the Beta family.

**Usage**

Beta(link = "logit")

**Arguments**

- **link**
  - the link function

**Value**

Output as for other GLM families

---

CAR  
*Conditional Autoregressive Family*

**Description**

A function used in the hglm package which extends the usage of the CAR family.

**Usage**

CAR(D, link = "identity", link.rand_disp = "inverse")

**Arguments**

- **D**
  - the D matrix of the Markov Random Field model.
- **link**
  - the link function for the random effects.
- **link.rand_disp**
  - the link function for the random effects dispersion parameter.
Value

Output specific for hglm fit, including eigen values and vectors of $D$.

References

Moudud Alam, Lars Ronnegard, Xia Shen (2014). **Fitting conditional and simultaneous autoregressive spatial models in hglm. Submitted.**

---

**hglm**

*Fitting Hierarchical Generalized Linear Models*

Description

hglm is used to fit hierarchical generalized linear models. It can be used for linear mixed models and generalized linear models with random effects for a variety of links and a variety of distributions for both the outcomes and the random effects. Fixed effects can also be fitted in the dispersion part of the model. The function can be called either by specifying the design matrices or as a **formula**.

Usage

```r
hglm(X = NULL, y = NULL, Z = NULL, family = gaussian(link = identity),
    rand.family = gaussian(link = identity), method = "EQL",
    conv = 1e-6, maxit = 50, startval = NULL, fixed = NULL,
    random = NULL, X.disp = NULL, disp = NULL, link.disp = "log",
    X.rand.disp = NULL, rand.disp = NULL, link.rand.disp = "log",
    data = NULL, weights = NULL, fix.disp = NULL, offset = NULL,
    RandC = ncol(Z), sparse = TRUE, vcovmat = FALSE,
    calc.like = FALSE, bigRR = FALSE, verbose = FALSE, ...)
```

Arguments

- **X**
  - `matrix`. The design matrix for the fixed effects.
- **y**
  - `numeric`. The dependent variable.
- **Z**
  - `matrix`. The design matrix for the random effects.
- **family**
  - `family`. The description of the error distribution and link function to be used in the mean part of the model. (See `family` for details of family functions.)
- **rand.family**
  - `family`. The description of the distribution and link function to be used for the random effect.
- **method**
  - `character`. Estimation method where EQL is the method of interconnected GLMs presented in Lee et al (2006). Apart from the default option EQL there is also an EQL1 option, which improves estimation for GLMMs (especially for Poisson models with a large number of levels in the random effects).
- **conv**
  - `numeric`. The convergence criteria (change in linear predictor between iterations).
- **maxit**
  - `numeric`. Maximum number of iterations in the hglm algorithm.
hglm

**startval** numeric. A vector of starting values in the following order: fixed effects, random effect, variance of random effects, variance of residuals.

**fixed** formula. A `formula` specifying the fixed effects part of the model.

**random** formula. A one-sided `formula` specifying the random effects part of the model.

**X.disp** matrix. The design matrix for the fixed effects in the residual dispersion part of the model.

**disp** formula. A one-sided `formula` specifying the fixed effects in the residual dispersion part of the model.

**link.disp** character. The link function for the residual dispersion part of the model.

**X.rand.disp** matrix. The design matrix for the fixed effects in the random effects dispersion part of the model.

**rand.disp** formula. A one-sided `formula` specifying the fixed effects in the random effects dispersion part of the model.

**link.rand.disp** character. The link function for the random effects dispersion part of the model.

**data** `data.frame`. The data frame to be used together with fixed and random.

**weights** numeric. Prior weights to be specified in weighted regression.

**fix.disp** numeric. A numeric value if the dispersion parameter of the mean model is known, e.g., 1 for binomial and Poisson model.

**offset** An offset for the linear predictor of the mean model.

**RandC** numeric. Integers (possibly a vector) specifying the number of column of Z to be used for each of the random-effect terms.

**sparse** logical. If TRUE, the computation is to be carried out by using sparse matrix technique.

**vcovmat** logical. If TRUE, the variance-covariance matrix is returned.

**calc.like** logical. If TRUE, likelihoods will be computed at convergence and will be shown via the print or summary methods on the output object.

**bigRR** logical. If TRUE, and only for the Gaussian model with one random effect term, a specific algorithm will be used for fast fitting high-dimensional (p \(\gg\) n) problems. See Shen et al. (2013) for more details of the method.

**verbose** logical. If TRUE, more information is printed during model fitting process.

... not used.

**Details**

Models for `hglm` are either specified symbolically using `formula` or by specifying the design matrices (`X`, `Z` and `X.disp`). Currently, only the extended quasi likelihood (EQL) method is available for the estimation of the model parameters. Only for the Gaussian-Gaussina linear mixed models, it is REML. It should be noted that the EQL estimator can be biased and inconsistent in some special cases e.g. binary pair matched response. A higher order correction might be useful to correct the bias of EQL (Lee et al. 2006). But, those corrections are not implemented in the current version. By default, the dispersion parameter is always estimated via EQL. If the dispersion parameter of the mean model is to be held constant, for example if it is desired to be 1 for binomial and Poisson family, then `fix.disp=value` where, value=1 for the above example, should be used.
Value

It returns an object of class `hglm` consisting of the following values.

- `fixef`: fixed effect estimates.
- `ranef`: random effect estimates.
- `RandC`: integers (possibly a vector) specified the number of column of Z to be used for each of the random-effect terms.
- `varFix`: dispersion parameter of the mean model (residual variance for LMM).
- `varRanef`: dispersion parameter of the random effects (variance of random effects for GLMM).
- `CAR.rho`: parameter estimate for a MRF spatial model.
- `CAR.tau`: parameter estimate for a MRF spatial model.
- `iter`: number of iterations used.
- `Converge`: specifies if the algorithm converged.
- `SeFe`: standard errors of fixed effects.
- `SeRe`: standard errors of random effects.
- `dfReFe`: deviance degrees of freedom for the mean part of the model.
- `SummVC1`: estimates and standard errors of the linear predictor in the dispersion model.
- `SummVC2`: estimates and standard errors of the linear predictor for the dispersion parameter of the random effects.
- `dev`: individual deviances for the mean part of the model.
- `hv`: hatvalues for the mean part of the model.
- `resid`: studentized residuals for the mean part of the model.
- `fv`: fitted values for the mean part of the model.
- `disp.fv`: fitted values for the dispersion part of the model.
- `disp.resid`: standardized deviance residuals for the dispersion part of the model.
- `link.disp`: link function for the dispersion part of the model.
- `vcov`: the variance-covariance matrix.
- `likelihood`: a list of log-likelihood values for model selection purposes, where `hlik` is \(-2\) times the log-h-likelihood, `pvh` \(-2\) times the adjusted profile log-likelihood profiled over random effects, `pbvh` \(-2\) times the adjusted profile log-likelihood profiled over fixed and random effects, and `cAIC` the conditional AIC.

Author(s)

Moudud Alam, Lars Ronnegard, Xia Shen
References


See Also

hglm2

Examples

# Find more examples and instructions in the package vignette:
# vignette('hglm')

require(hglm)

# ------------------------- #
# semiconductor example #
# ------------------------- #

data(semiconductor)

m11 <- hglm(fixed = y ~ x1 + x3 + x5 + x6,
            random = ~ 1|Device,
            family = Gamma(link = log),
            disp = ~ x2 + x3, data = semiconductor)

summary(m11)

plot(m11, cex = .6, pch = 1,
     cex.axis = 1/.6, cex.lab = 1/.6,
     cex.main = 1/.6, mar = c(3, 4.5, 0, 1.5))

# ------------------------- #
# redo it using hglm2 #
# ------------------------- #

m12 <- hglm2(y ~ x1 + x3 + x5 + x6 + (1|Device),
              family = Gamma(link = log),
              disp = ~ x2 + x3, data = semiconductor)

summary(m12)
# redo it using matrix input #

attach(semiconductor)
m13 <- hglm(y = y, X = model.matrix(~ x1 + x3 + x5 + x6),
  Z = kronecker(diag(16), rep(1, 4)),
  X.disp = model.matrix(~ x2 + x3),
  family = Gamma(link = log))
summary(m13)

# verbose & likelihoods #

m14 <- hglm(fixed = y ~ x1 + x3 + x5 + x6,
  random = ~ 1|Device,
  family = Gamma(link = log),
  disp = ~ x2 + x3, data = semiconductor,
  verbose = TRUE, calc.like = TRUE)
summary(m14)

# simulated example with 2 random effects terms #

## Not run:
set.seed(911)
x1 <- rnorm(100)
x2 <- rnorm(100)
x3 <- rnorm(100)
z1 <- factor(rep(LETTERS[1:10], rep(10, 10)))
z2 <- factor(rep(letters[1:5], rep(20, 5)))
Z1 <- model.matrix(~ 0 + z1)
Z2 <- model.matrix(~ 0 + z2)
u1 <- rnorm(10, 0, sqrt(2))
u2 <- rnorm(5, 0, sqrt(3))
y <- 1 + 2*x1 + 3*x2 + Z1*u1 + Z2*u2 + rnorm(100, 0, sqrt(exp(x3)))
dd <- data.frame(x1 = x1, x2 = x2, x3 = x3, z1 = z1, z2 = z2, y = y)

(m21 <- hglm(X = cbind(rep(1, 100), x1, x2), y = y, Z = cbind(Z1, Z2),
  RandC = c(10, 5)))
summary(m21)
plot(m21)

# m21 is the same as:
(m21b <- hglm(X = cbind(rep(1, 100), x1, x2), y = y, Z = cbind(Z1, Z2),
  rand.family = list(gaussian(), gaussian()), RandC = c(10, 5))

(m22 <- hglm2(y ~ x1 + x2 + (1|z1) + (1|z2), data = dd, vcovmat = TRUE))
image(m22$vcov, main = 'Variance-covariance Matrix')
summary(m22)
plot(m22)
**hglm2**

**Fitting Hierarchical Generalized Linear Models**

**Description**

`hglm2` is used to fit hierarchical generalized linear models. It extends the `hglm` function by allowing for several random effects, where the model is specified in `lme` convension, and also by implementing sparse matrix techniques using the Matrix library.

**Usage**

```r
hglm2(meanmodel, data = NULL, family = gaussian(link = identity),
      rand.family = gaussian(link = identity), method = "EQL",
      conv = 1e-6, maxit = 50, startval = NULL,
      X.disp = NULL, disp = NULL, link.disp = "log",
      weights = NULL, fix.disp = NULL, offset = NULL,
      ...)```

```r
m31 <- hglm2(y ~ x1 + x2 + (1|z1) + (1|z2), disp = ~ x3, data = dd)
print(m31)
summary(m31)
plot(m31)

# Markov random field (MRF) model
# -----------------------------
data(cancer)
logE <- log(E)
X11 <- model.matrix(~Pa)
m41 <- hglm(X = X11, y = 0, Z = diag(length(0)),
           family = poisson(), rand.family = CAR(D = nbr),
           offset = logE, conv = 1e-9, maxit = 200, fix.disp = 1)
summary(m41)

data(ohio)
m42 <- hglm(fixed = Medianscore - 1,
           random = ~ 1 | district,
           rand.family = CAR(D = ohioDistrictDistMat),
           data = ohioMedian)
summary(m42)
require(sp)
districtShape <- as.numeric(substr(as.character(ohioShape@data$UNSIDDP), 3, 7))
CARfit <- matrix(m42$ranef + m42$fixef, dimnames = list(rownames(ohioDistrictDistMat), NULL))
spplot(ohioShape, zcol = "CAR", main = "Fitted values from CAR",
col.regions = heat.colors(1000)[1000:1], cuts = 1000)
```

```r
```
sparse = TRUE, vcovmat = FALSE, calc.like = FALSE,
bigRR = FALSE, verbose = FALSE, ...)

Arguments

meanmodel formula. A two sided formula specifying the fixed and random terms in lme4
c convention, e.g. y ~ x1 + (1|id) indicates y as response, x1 as the fixed effect
and (1|id) represent a random intercept for each level of id.
data data.frame. An optional data frame from where the variables in the meanmodel
(and possibly disp) are to be obtained. It is expected that the data frame does
not contain any missing value.
family family. The description of the error distribution and link function to be used in
the mean part of the model. (See family for details of family functions.)
rand.family family. The description of the distribution and link function to be used for the
random effect.
method character. Estimation method where EQL is the method of interconnected
GLMs presented in Lee et al (2006). Apart from the default option EQL there
is also an EQL1 option, which improves estimation for GLMMs (especially for
Poisson models with a large number of levels in the random effects).
conv numeric. The convergence criteria (change in linear predictor between itera-
tions).
maxit numeric. Maximum number of iterations in the hglm algorithm.
startval numeric. A vector of starting values in the following order: fixed effects, ran-
dom effect, variance of random effects, variance of residuals.
X.disp matrix. The design matrix for the fixed effects in the dispersion part of the
model.
disp formula. A one-sided formula specifying the fixed effects in the dispersion part of the
model.
link.disp character. The link function for the dispersion part of the model.
weights numeric. Prior weights to be specified in weighted regression.
fix.disp numeric. A numeric value if the dispersion parameter of the mean model is
known, e.g., 1 for binomial and Poisson model.
offset An offset for the linear predictor of the mean model.
sparse logical. If TRUE, the computation is to be carried out by using sparse matrix
technique.
vcovmat logical. If TRUE, the variance-covariance matrix is exported.
calc.like logical. If TRUE, likelihoods will be computed at convergence and will be shown
via the print or summary methods on the output object.
bigRR logical. If TRUE, and only for the Gaussian model with one random effect term, a
specific algorithm will be used for fast fitting high-dimensional (p > n) problems.
See Shen et al. (2013) for more details of the method.
verbose logical. If TRUE, more information is printed during model fitting process.
... not used.
Details

Models for hglm are either specified symbolically using `formula` or by specifying the design matrices (X, Z and X.disp). Currently, only the extended quasi likelihood (EQL) method is available for the estimation of the model parameters. Only for the Gaussian-Gaussian linear mixed models, it is REML. It should be noted that the EQL estimator can be biased and inconsistent in some special cases e.g. binary pair matched response. A higher order correction might be useful to correct the bias of EQL (Lee et al. 2006). But, those corrections are not implemented in the current version. By default, the dispersion parameter is always estimated via EQL. If the dispersion parameter of the mean model is to be held constant, for example if it is desired to be 1 for binomial and Poisson family, then `fix.disp=value` where, value=1 for the above example, should be used.

Value

It returns an object of class `hglm` consisting of the following values.

- `fixef` fixed effect estimates.
- `ranef` random effect estimates.
- `RandC` integers (possibly a vector) specified the number of column of Z to be used for each of the random-effect terms.
- `varFix` dispersion parameter of the mean model (residual variance for LMM).
- `varRanef` dispersion parameter of the random effects (variance of random effects for GLMM).
- `iter` number of iterations used.
- `Converge` specifies if the algorithm converged.
- `SeFe` standard errors of fixed effects.
- `SeRe` standard errors of random effects.
- `dfReFe` deviance degrees of freedom for the mean part of the model.
- `SummVC1` estimates and standard errors of the linear predictor in the dispersion model.
- `SummVC2` estimates and standard errors of the linear predictor for the dispersion parameter of the random effects.
- `dev` individual deviances for the mean part of the model.
- `hv` hatvalues for the mean part of the model.
- `resid` studentized residuals for the mean part of the model.
- `fv` fitted values for the mean part of the model.
- `disp.fv` fitted values for the dispersion part of the model.
- `disp.resid` standardized deviance residuals for the dispersion part of the model.
- `link.dispatch` link function for the dispersion part of the model.
- `vcov` the variance-covariance matrix.
- `likelihood` a list of log-likelihood values for model selection purposes, where `$hlik` is - 2 times the log-h-likelihood, `$pvh` -2 times the adjusted profile log-likelihood profiled over random effects, `$pbvh` -2 times the adjusted profile log-likelihood profiled over fixed and random effects, and `$cAIC` the conditional AIC.
Author(s)
Moudud Alam, Xia Shen, Lars Ronnegard

References


See Also
hglm

Examples

# Find more examples and instructions in the package vignette:
# vignette('hglm')

require(hglm)

# ---------------------------- #
# semiconductor example #
# ---------------------------- #

data(semiconductor)

m11 <- hglm(fixed = y ~ x1 + x3 + x5 + x6,
            random = ~ 1|Device,
            family = Gamma(link = log),
            disp = ~ x2 + x3, data = semiconductor)

summary(m11)
plot(m11, cex = .6, pch = 1,
     cex.axis = 1/.6, cex.lab = 1/.6,
     cex.main = 1/.6, mar = c(3, 4.5, 0, 1.5))

# ---------------------------- #
# redo it using hglm2 #
# ---------------------------- #
m12 <- hglm2(y ~ x1 + x3 + x5 + x6 + (1|Device),
  family = Gamma(link = log),
  disp = ~ x2 + x3, data = semiconductor)
summary(m12)

# ---------------------------- #
# redo it using matrix input #
# ---------------------------- #

attach(semiconductor)

m13 <- hglm(y = y, X = model.matrix(~ x1 + x3 + x5 + x6),
  Z = kronecker(diag(16), rep(1, 4)),
  X.disp = model.matrix(~ x2 + x3),
  family = Gamma(link = log))
summary(m13)

# ---------------------------- #
# verbose & likelihoods #
# ---------------------------- #

m14 <- hglm(fixed = y ~ x1 + x3 + x5 + x6,
  random = ~ 1|Device,
  family = Gamma(link = log),
  disp = ~ x2 + x3, data = semiconductor,
  verbose = TRUE, calc.like = TRUE)
summary(m14)

# ------------------------------- #
# simulated example with 2 random effects terms #
# ----------------------------------------------- #

## Not run:
set.seed(911)
x1 <- rnorm(100)
x2 <- rnorm(100)
x3 <- rnorm(100)
z1 <- factor(rep(LETTERS[1:10], rep(10, 10)))
z2 <- factor(rep(letters[1:5], rep(20, 5)))
Z1 <- model.matrix(~ 0 + z1)
Z2 <- model.matrix(~ 0 + z2)
u1 <- rnorm(100, 0, sqrt(2))
u2 <- rnorm(50, 0, sqrt(3))
y <- 1 + 2*x1 + 3*x2 + Z1%*%u1 + Z2%*%u2 + rnorm(100, 0, sqrt(exp(x3)))
dd <- data.frame(x1 = x1, x2 = x2, x3 = x3, z1 = z1, z2 = z2, y = y)

(m21 <- hglm(X = cbind(rep(1, 100), x1, x2), y = y, Z = cbind(Z1, Z2),
  RandC = c(10, 5)))
summary(m21)
plot(m21)

(m22 <- hglm2(y ~ x1 + x2 + (1|z1) + (1|z2), data = dd, vcovmat = TRUE))
image(m22$vcov, main = 'Variance-covariance Matrix')
summary(m22)
plot(m22)
lrt <- hglm2(y ~ x1 + x2 + (1|z1) + (1|z2), disp = ~ x3, data = dd)
print(m31)
summary(m31)
plot(m31)

## End(Not run)

---

**lrt**  
*Likelihood-ratio test for variance components in hglm*

---

**Description**

Likelihood-ratio test for the estimated variance components (or other dispersion parameters) in hglm.

**Usage**

```r
lrt(hglm.obj1, hglm.obj2 = NULL)
```

**Arguments**

- `hglm.obj1`: a fitted hglm object.
- `hglm.obj2`: optional, another fitted hglm object to be tested against hglm.obj1.

**Details**

When `hglm.obj2` = NULL, all the random effects variance components in hglm.obj1 are tested against the null model with only fixed effects. The degree of freedom is determined by comparing the number of random effects terms in hglm.obj1 and hglm.obj2 or the null fixed-effects-only model. Note that the likelihood-ratio test statistic for variance estimates, which are bounded above zero, follows a 50:50 mixture distribution of chi-square with 0 and 1 degree of freedom (Self and Liang 1987 JASA).

**Value**

Printout summary of the likelihood-ratio test results. Test statistic, p-value, etc. are returned.

**References**

Examples

```r
require(hglm)

## Not run:
set.seed(911)
x1 <- rnorm(100)
x2 <- rnorm(100)
x3 <- rnorm(100)
z1 <- factor(rep(LETTERS[1:10], rep(10, 10)))
z2 <- factor(rep(letters[1:5], rep(20, 5)))
Z1 <- model.matrix(~ 0 + z1)
Z2 <- model.matrix(~ 0 + z2)
u1 <- rnorm(10, 0, sqrt(2))
u2 <- rnorm(5, 0, sqrt(3))
y <- 1 + 2*x1 + 3*x2 + Z1%*%u1 + Z2%*%u2 + rnorm(100, 0, sqrt(exp(x3)))
dd <- data.frame(x1 = x1, x2 = x2, x3 = x3, z1 = z1, z2 = z2, y = y)

m20 <- hglm(x = cbind(rep(1, 100), x1, x2), y = y, Z = Z1,
            calc.like = TRUE)

lrt(m20)

m21 <- hglm(x = cbind(rep(1, 100), x1, x2), y = y, Z = cbind(Z1, Z2),
            RandC = c(10, 5), calc.like = TRUE)

lrt(m20, m21)

## End(Not run)
```

Description

Plots residuals for the mean and dispersion models, individual deviances and hatvalues for `hglm` objects

Usage

```r
## S3 method for class 'hglm'
plot(x, pch = "+", pcol = 'slateblue', lcol = 2,
     device = NULL, name = NULL, ...)
```

Arguments

- `x`: the `hglm` object to be plotted
- `pch`: symbol used in the plots
pcol  color of points
lcol  color of lines
device if NULL, plot on screen devices, if 'pdf', plot to PDF files in the current working directory.
name  a string gives the main name of the PDF file when device = 'pdf'.
      graphical parameters

Details
A S3 generic plot method for hglm objects. It produces a set of diagnostic plots for a hierarchical model.

Author(s)
Xia Shen

Examples

# ------------------------ #
# semiconductor example #
# ------------------------ #
data(semiconductor)

h.gamma.normal <- hglm(fixed = y ~ x1 + x3 + x5 + x6,
                        random = ~ 1|Device,
                        family = Gamma(link = log),
                        disp = ~ x2 + x3, data = semiconductor)

summary(h.gamma.normal)
plot(h.gamma.normal, cex = .6, pch = 1,
     cex.axis = 1/6, cex.lab = 1/6,
     cex.main = 1/6, mar = c(3, 4.5, 0, 1.5))

# ------------------------ #
# redo it using hglm2 #
# ------------------------ #

m1 <- hglm2(y ~ x1 + x3 + x5 + x6 + (1|Device),
            family = Gamma(link = log),
            disp = ~ x2 + x3, data = semiconductor)

summary(m1)
plot(m1, cex = .6, pch = 1,
     cex.axis = 1/6, cex.lab = 1/6,
     cex.main = 1/6, mar = c(3, 4.5, 0, 1.5))

# ----------------------------------------------- #
# simulated example with 2 random effects terms #
# ----------------------------------------------- #
## Not run:
set.seed(911)
x1 <- rnorm(100)
**SAR**  

*Simultaneous Autoregressive Family*

**Description**

A function used in the `hglm` package which extends the usage of the SAR family.

**Usage**

```r
SAR(D, link = "identity", link.rand.disp = "inverse.sqrt")
```

**Arguments**

- `D`  
  the D matrix of the SAR model.

- `link`  
  the link function for the random effects.

- `link.rand.disp`  
  the link function for the random effects dispersion parameter.

**Value**

Output specific for `hglm` fit, including eigen values and vectors of D.
References

Index

*Topic package
  hglm-package, 2

Beta, 3

CAR, 3

family, 4, 10
formula, 2, 4, 5, 10, 11

hglm, 3, 4, 12
hglm-package, 2
hglm2, 3, 7, 9

lrt, 14

plot(plot.hglm), 15
plot.hglm, 3, 15

SAR, 17